

Utveckling och lansering av en Android-applikation

Toni Malila

Examensarbete
Informations- och medieteknik
2013

Toni Malila

EXAMENSARBETE	
Arcada	
Utbildningsprogram:	Informations- och medieteknik
Identifikationsnummer:	
Författare:	Toni Malila
Arbetets namn:	Utveckling och lansering av en Android-applikation
Handledare (Arcada):	
Uppdragsgivare:	
<p>Sammandrag:</p> <p>Arbetet handlar om utvecklandet av en Android-applikation, lanseringen av applikationen på Google Play och en beskrivning över hur man kan öka nerladdningar för den lanserade applikationen. Applikationen som utvecklas är en stämpelklocka där användare med NFC-taggar kan identifiera sig och logga sina arbets- eller projekttimmar. Arbetet består av fyra delar, en teoretisk del där de använda verktygen och teknologierna presenteras kort och förklarar deras anknytning till utvecklingen. I andra delen presenteras lanseringsprocessen ur en allmän synvinkel med Google Plays riktlinjer. Den tredje delen består av utvecklingsprocessen för applikationen. Där presenteras applikationens funktionsprincip och lösningar för de viktigaste funktionerna i applikationen. I fjärde delen presenteras lanseringen av den utvecklade applikationen. I slutet av arbetet tas det upp några metoder för att öka nerladdningar för den lanserade applikationen, samt presenteras nerladdnings statistik för den utvecklade applikationen.</p>	
Nyckelord:	Android,NFC,Google Play, lansering, utveckling
Sidantal:	
Språk:	
Datum för godkännande:	

DEGREE THESIS	
Arcada	
Degree Programme:	Information and Media Technology
Identification number:	
Author:	Toni Malila
Title:	Development and launching of an Android application
Supervisor (Arcada):	
Commissioned by:	
<p>Abstract:</p> <p>The thesis is about the development of an Android application, launching of the application on Google Play and a description of how to increase the number of downloads for the launched application. The developed application is a punch clock where users can use NFC-tags to identify themselves and log time used on projects or at work. The thesis consists of four parts. A theoretical part that consists of a presentation of the technologies and tools used in the project, and what parts they played in the project. In the second part the launch of an application on Google Play will be presented from a general perspective using the guidelines of Google Play. The third part consists of the development of the application, where key elements in the development will be presented. In the fourth part the launch of the developed application will be presented. At the end of the project some methods to increase the number of downloads for the application will be presented along with some download statistics for the application developed in this project.</p>	
Keywords:	Android,NFC,Google Play, launching, development
Number of pages:	
Language:	Swedish
Date of acceptance:	

OPINNÄYTE	
Arcada	
Koulutusohjelma:	Tieto ja Mediatekniikka
Tunnistenumero:	
Tekijä:	Toni Malila
Työn nimi:	Android-sovelluksen kehitys ja julkaisu
Työn ohjaaja (Arcada):	
Toimeksiantaja:	
<p>Tiivistelmä:</p> <p>Työ kertoo Android-sovelluksen kehityksestä, sovelluksen julkaisemisesta Google Play:ssä ja selvitys miten julkaistulle sovellukselle voi saada enemmän latauksia. Kehitetty sovellus on kellokortti-sovellus jossa käyttäjät voivat tunnistautua NFC-korteilla ja kirjata työtuntinsa tai projektissa käyttämänsä ajan. Työ koostuu neljästä osasta. Ensimmäisessä osassa esitetään työkalut ja teknologiat joita käytettiin sovelluksen kehityksessä. Toisessa osassa esitetään sovelluksen julkaisu prosessi yleisestä näkökulmasta, käyttäen Google Play:n ohjeita. Kolmannessa osassa käsitellään sovelluksen kehitysvaihe. Tässä osassa esitellään sovelluksen toimintaperiaate sekä käydään läpi tärkeimpien toimintojen ratkaisutavat. Neljännessä osassa esitellään kehitetyn sovelluksen julkaisu Google Play:ssä. Työn lopussa esitellään menetelmiä joilla voi saada enemmän latauksia jo julkaistulle sovellukselle, sekä esitellään kehitetyn sovelluksen lataus statistiikkaa.</p>	
Avainsanat:	Android,NFC,Google Play, lanseeraus, kehitys
Sivumäärä:	
Kieli:	Ruotsi
Hyväksymispäivämäärä:	

INNEHÅLL

1	Inledning.....	10
2	Lanseringsmetoder	11
2.1	Förberedning för lansering	11
2.1.1	<i>Introduktion</i>	<i>11</i>
2.1.2	<i>Förberedande av material och andra resurser</i>	<i>13</i>
2.1.3	<i>Förberedande av applikationen</i>	<i>14</i>
2.2	Lansering av applikationen.....	15
2.2.1	<i>Byggande av lanseringsfärdiga applikationen</i>	<i>15</i>
2.2.2	<i>Förberedande och testning av applikationen</i>	<i>16</i>
2.2.3	<i>Lansering på Google Play</i>	<i>16</i>
2.2.4	<i>Utdelning via epost</i>	<i>17</i>
2.2.5	<i>Utdelning på webbsida</i>	<i>18</i>
3	Utvecklingsteknologier	18
3.1	Android	18
3.2	NFC	19
3.3	SQL	19
3.3.1	<i>Allmänt om SQL</i>	<i>19</i>
3.3.2	<i>SQLite.....</i>	<i>20</i>
3.4	Eclipse	21
3.5	Java	21
4	Utvecklingsprocessen.....	22
4.1	Planering och kravställning	22
4.2	Användargränssnittet	23
4.3	Lösning av kravställningarna.....	26
4.3.1	<i>Användarhantering i databas</i>	<i>27</i>
4.3.2	<i>Skrivande och modifierande av NFC-taggar</i>	<i>27</i>
4.3.3	<i>"Användare kan identifieras av applikationen med NFC-taggar"</i>	<i>28</i>
4.3.4	<i>Användare kan logga in, till lunch och ut.....</i>	<i>29</i>
4.3.5	<i>Exportering av databaserna</i>	<i>30</i>
5	Lanseringsprocessen.....	30
5.1	Förberedningar	31
5.2	Kompilering och signering	32
5.3	Google Play lansering	32
6	Hur öka nerladdningar.....	33

6.1	Metoder	33
6.1.1	<i>Applikationen</i>	34
6.1.2	<i>Användarvärderingar</i>	35
6.1.3	<i>Marknadsföring</i>	35
6.2	Process	35
6.3	Resultat och analys	36
7	slutsatser	38
	Källor	39
	Bilagor	42

Figurer

Figur 1. De olika stegen i förberedning för lansering	12
Figur 2. Diagram över applikationens användargränssnitt.....	24
Figur 3. Huvudmeny och användaraktivitet vyn av applikationen.....	25
Figur 4. Användar-och databashanteringsvyn från applikationen.....	26
Figur 5. Användardatabas exempel	27
Figur 6. Funktionen ”onNewIntent”	29
Figur 7. Loggdatabas exempel.....	30
Figur 8. Totala installeringar	36
Figur 9. Aktiva installeringar	37

Tabeller

Tabell 1. Förhållande mellan ikon storlek och skärmdensitet	31
---	----

FÖRKORTNINGAR

SDK	Software Development Kit
API	Application Programming Interface
NDK	Native Development Kit
NFC	Near Field Communication
RFID	Radio Frequency Identification
SQL	Structured Query Language
ANSI	American National Standards Institute
ISO	International Organization for Standardization
XML	Extensible Markup Language
IDE	Integrated Development Environment
ADT	Android Development Tools
JVM	Java Virtual Machine
APK	Application Package File
RAM	Random Access Memory
EULA	End User License Agreement
JDK	Java Development Kit
CSV	Comma-Separated Values
CA	Certificate Author

1 INLEDNING

Arbetet handlar om utvecklingen av en Android-applikation, lanseringen av den på Google Play och metoder för att få bättre synlighet och därmed mera nerladdningar för applikationen. I dagens läge när hundratals applikationer lanseras dagligen på webbbaffärer är det lätt att egna applikationer blir borttappade mellan massorna. I arbetet presenteras några metoder för att se till att applikationen inte blir bortglömd och istället får sin andel av nerladdningar.

Syftet med projektet är att klargöra lanseringsprocessen av en Android-applikation på Google Play, samt att presentera metoder för att öka antalet nerladdningar för applikationen. I arbetet beskrivs i stora drag utvecklandet av en Android-applikation som sedan lanseras som ett praktiskt exempel. Applikationens kravförställningar presenteras och hur dessa är lösta i utvecklingsskedet, samt presenteras en diagram över användargränssnittet. I kapitlet Lanseringsprocessen beskrivs steg för steg hur lanseringsprocessen gick till för applikationen som utvecklades för det här arbetet. Dessutom beskrivs lanseringsprocessen ur en allmän synvinkel enligt Google Plays riktlinjer i kapitlet Lanseringsmetoder. Där beskrivs vilka steg bör tas när man förbereder sig för lanseringen av applikationen som t.ex. förberedande av marknadsföringsmaterial och en ikon för applikationen, samt förberedande av applikationen till ett format som kan sedan lanseras på Google Play och via andra källor. I slutet av arbetet presenteras några metoder för att få mera nerladdningar för en lanserad applikation. Metoderna är t.ex. att marknadsföra applikationen på sociala medier samt få värderingar för applikationen, vilket förklaras noggrannare i nämnda kapitlet. Fokus av arbetet är att klargöra lanseringsprocessen så att läsaren får en bild av vad det egentligen går ut på och hur man kan därmed förbereda sig för lanseringen redan vid utvecklingsskedet och att förstå hur Google Play indexerar och väljer att visa applikationer, samt hur detta kan påverkas.

Applikationen som utvecklas och lanseras fungerar som en stämpelklocka som använder sig av NFC-taggar för att identifiera olika användare. NFC-taggar är små enheter som t.ex. klistermärken eller kort med inbyggda mikrochips som kan med hjälp av trådlös teknologi spara små mängder data i sig själva. Användarna kan efter identifieringen göra en loggning för när de kommer till arbetet, går på lunch samt återvänder från lunch

och loggar ut. Loggningarna och användarna sparas i lokala databaser som kan exporteras till csv-filer som är enkelt läsbara med datorer.

Arbetet fokuserar inte på marknadsförningsdelen av applikationen, fastän metoderna som presenteras för att få mera nerladdningar kan ses som marknadsföring.

2 LANSERINGSMETODER

I detta kapitel presenteras lanseringsprocessen ur en allmän synvinkel där de olika delmomenten och möjligheterna för lansering av en Android-applikation beskrivs oberoende av applikationens typ. I nästa kapitel går igenom hur applikationen som utvecklades för detta arbete lanserades i praktiken.

Lanseringen av en applikation kan delas in i två huvuddelar, förberedelse av applikationen för lansering och lansering av applikationen. I förberedningsdelen kompileras en färdig applikation som användare kan ladda ner och installera på sina Android-enheter. I lanseringsdelen släpps applikationen till allmänheten, säljs och distribueras till användare. (*Publishing Overview*)

2.1 Förberedning för lansering

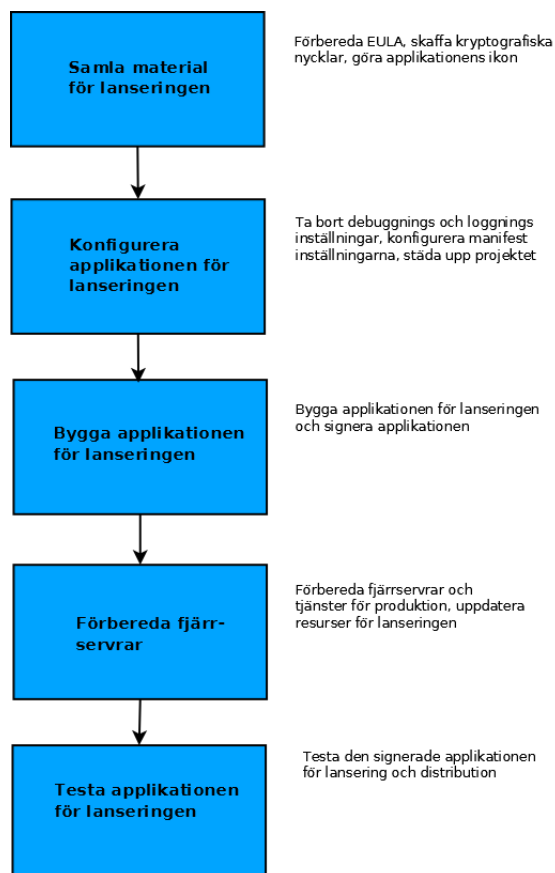
I det här avsnittet går förberedningar igenom för att få en lanseringsfärdig applikation i .apk-format(Application package file) som kan uppladdas till en applikationsaffär som Google Play eller utdelas direkt till användare.

2.1.1 Introduktion

För att kunna lansera applikationen till användare måste ett lanseringsfärdigt paket byggas av applikationen som användare kan installera och köra på sina Android-enheter. Paketet innehåller kompilerad källkod, resurser och manifest fil, som är en fil

som presenterar nödvändig information om applikationen till Android-systemet. Paketet innehåller alltså samma data som avlusningspaketet och båda paketen byggs upp med samma programvara. Skillnaden mellan avlusningspaketet och lanseringsfärdiga-paketet är att lanseringsfärdiga-paketets .apk-fil är signerat med utvecklarens egna certifikat och optimerad med zipalign-verktyget. Zipalign-verktyget gör att all okomprimerad data som t.ex. bilder eller raw-filer är grupperade i 4-bytes grupperingar. Det möjliggör att alla delar kan åtkommas direkt med mmap()-funktionen även om de innehåller binär data med grupperings begränsningar. Mmap()-funktionen är ett Unix system anrop som beskriver filer och enheternas placering i minnet. Fördelen med detta är en minskning i mängden RAM(Random Access Memory) som används när applikationen körs.(zipalign)

Förberedningen kan indelas i fem olika huvudsteg enligt Googles rekommendationer varav varje innebär några mindre arbeten beroende på hur applikationen kommer att utdelas, stegen kan ses i figur 1 nedan.(Preparing for Release)



Figur 1. De olika stegen i förberedning för lansering (Preparing for Release)

2.1.2 Förberedande av material och andra resurser

För att lansera en applikation behöver man åtminstone ett certifierat nyckelpar för signering av applikationen och en ikon för applikationen. Nedan presenteras dessa i mer detalj.

Android kräver att varje applikation är digitalt signerad med utvecklarens privata nyckel. Med hjälp av certifikatet kan man sedan identifiera utvecklaren för Googles tjänster och därmed bygga ett förtroende mellan utvecklaren och Google, samt dess applikationer. Med förtroende menas att utvecklaren kan identifiera sig med Google som applikationens riktiga ägare och utvecklare och kan därmed använda sig av Googles olika tjänster. Certifikatet som man använder behöver inte vara signerad av en utomstående certifieringsauktoritet, utan det kan vara självsignerat. Google rekommenderar att certifikatet är i kraft längre än förväntade livslängden för applikationen men rekommenderas vara åtminstone 25 år. För att certifikatet skall godkännas bör det dessutom ha ett utgångsdatum senare än 22 oktober 2033, i andra fall kan applikationen som signerats ~~med nyckeln~~ inte uppladdas till Google Play. Det godkänns inte heller att applikationen signeras med en avlusningsnyckel som automatiskt genereras när applikationen utvecklas med Eclipse ADT. (*Signing your Applications*)

Ifall applikationen använder sig av vissa tredje parts tjänster eller regeluppsättningar kan en extra nyckel krävas. Den nya nyckeln innehåller paketnamnet av applikationen tillsammans med SHA1 fingeravtrycket av utvecklarens nyckel som använts för att signera applikationen. Ett exempel är ifall applikationen använder sig av MapView-klassen som är en del av Google Maps externa regeluppsättningar, då måste applikationen registreras med Google Maps tjänsten för att erhålla en Maps API key.

Applikationen behöver en ikon som används för att identifiera applikationen på enheten för att kunna startas. Dessutom är den synlig på olika ställen på Android-enheten som inställningarna för alla applikationer, enhetens nerladdningar och i Google Play för att visa någonting fysiskt åt användare för att representera applikationen. Ikonen kan vara i flera olika storlekar beroende på enheten den är utvecklad för, men utöver dessa storlekar som beror på applikationen måste man ha en version med storleken 512x512 pixel

som används av Google Play på olika ställen, som på webbsidan av Google Play applikationen.

Google rekommenderar att man använder en EULA(End User Licence Agreement) för applikationen. EULA:n är ett juridiskt avtal mellan användaren och utvecklaren som specificerar hur applikationen får och inte får användas. Den kan hjälpa stöda utvecklarens rättigheter och dess immaterialrätt. EULA:n är inte en standard utan bör skrivas själv eller använda sig av färdiga modeller som kan hittas på nätet. (*EULA*)

Beroende på hur applikationen lanseras behövs dessutom marknadsförnings material för att stöda applikationen genom att beskriva den för användaren och dessutom ge en bild av applikationen utan att användaren behöver ladda ner den. Ifall applikationen publiceras på Google Play krävs produktinformation och åtminstone två skärmbilder av applikationen. Man kan även ladda upp olika upplösningars reklam eller en videolänk som visas på Google Play. (*Graphic and Image Assets*) (*Preparing for Release*)

2.1.3 Förberedande av applikationen

När man har förberett allt material från föregående avsnitt, kan man börja förberedandet av applikationens källkod och andra resurser för lansering.

Om man inte redan valt ett lämpligt paketnamn för applikationen, skall man göra det nu. Namnet kan inte ändras efter att det getts ut åt användare och skall därför vara lämpligt för hela applikationens livslängd. Namnet skall vara unikt eftersom det används för att identifiera applikationen. Vanligtvis används Javas namngivningspolicy, com.domain.app, där ”com” tyder på en kommersiell produkt, ”domain” är vanligen namnet på företaget och ”app” beskriver applikationen.

Applikationens loggningsfunktion skall avstängas för att applikationen skall kunna lanseras. Alla logfiler och möjliga statiska test filer skall också tas bort från projektet.

Alla onödiga filer skall tas bort eftersom de kan förhindra kompileringen av applikationen och göra att den beter sig oförväntat.

Manifest-filen skall kollas att den innehåller alla använda tjänster och alla oanvända ska tas bort. Dessutom skall man se till att Android-operativsystem versionerna är rätt definierade i manifest-filen ifall applikationen innehåller någon funktionalitet som inte finns i alla Android-versionerna.

Ifall applikationen använder sig av anslutningar till servrar eller andra fjärrtjänster skall man se till att adresserna till dessa är de som skall användas i slutmiljön.

Ifall applikationen kommer att utdelas via Google Play kan man sätta till Google Play Licensing till applikationen. Det ger utvecklaren kontroll över applikationen efter att användaren köpt den så att man kan t.ex. avgränsa viss funktionalitet för en bestämd tid för marknadsförnings syften. Vanligtvis används Licensing för att ge ut en gratis version av en vanligen avgiftsbelagd applikation, och m.h.a Licensing begränsas funktionaliteten av applikationen så att den inte motsvarar den betalda versionen. (*Preparing for Release*) Licensing fungerar på så sätt att när applikationen startas förfrågar den Google Play licens statuset för den nuvarande användaren och kan därmed tillåta eller neka åtkomst till funktionalitet av applikationen. (*Application Licensing*)

2.2 Lansering av applikationen

När applikationen är färdigt konfigurerad skall den byggas till en lanseringsfärdig apk-fil som är signerad och optimerad. För detta kan man använda Java Development Kit(JDK) som har verktygen Keytool och Jarsigner som signerar filen. Och därefter för kompilering och optimering av filen kan man använda Android SDK:s funktioner.

2.2.1 Byggnad av lanseringsfärdiga applikationen

Man kan göra hela processen med Eclipse ADT. Den har en inbyggd funktion som heter "Export Wizard" vilken steg för steg kompilerar applikationen, signerar den med användarens privata nyckel vid behov skapas även ett nyckelpar och optimerar applikationen med "zipalign"-verktyget. Förutom Eclipse ADT kan man också göra processen med "Ant"-skriptet, som är en Java-kompilator som fungerar med en kommandotolk, som är inkluderat i Android SDK:n. Skriptet utför samma process och ger samma resultat som Eclipse ADT.

2.2.2 Förberedande och testning av applikationen

När applikationen är i lanseringsformat rekommenderar Google att den installeras och testkörs på en fysisk enhet med slutmiljöns nätverks möjligheter och möjliga andra tjänster. Det innebär att kolla möjliga servrar och deras säkerhet, prestanda och konfigurationer och att all data på servern är tidsenlig. Dessutom rekommenderas det att applikationen körs åtminstone på en smarttelefon och en pekplatta för att se att användargränssnittet fungerar på båda och att både prestandan och batteriets livslängd är tillfredsställande.

Google rekommenderar att vid testning av applikationen på en fysisk enhet att prova ha enheten i både horisontellt läge och vertikalt läge, vilket innebär att luta enheten 90 grader så att skärmen anses vara "horisontellt" och se att allting fortfarande ritas rätt på skärmen. Dessutom skall man se till att applikationen hålls i samma vy under ändringen mellan de två olika lägen. Ifall applikationen är beroende av nätverksanslutning, SMS, Bluetooth eller GPS borde applikationen testas med dessa avstängda för att se hur den reagerar. (*What To Test*)

2.2.3 Lansering på Google Play

Om man väljer att lansera en applikation på Google Play får man tillgång till flera verktyg som erbjuds av Googles utvecklare webbsida. Verktygen innebär funktioner för analys av applikationens försäljning, identifiering av trender på marknaden, kontroll över

till vem applikationen är tillgänglig för och tjänster för att öka inkomsten av applikationen som applikations lisensering och ”in-app billing” som gör det möjligt att dela ut betalda tjänster inom applikationen. Vissa spel använder dessa funktioner för att sälja t.ex. mera nivåer som gör spelet på ett sätt ”längre”.

Lansering av en applikation på Google Play kan ses som en process med tre steg. I första steget förbereds marknadsförnings material som presenteras i föregående avsnitt. Under andra delen kan man konfigurera i vilka länder man vill att applikationen är tillgänglig, för vilka språk och hur mycket den kostar i de olika länderna ifall man väljer att ha ett pris på applikationen. Man kan välja att lägga till information om applikationen som typen av applikation, kategori och innehålls klassificering. Typen av applikation är antingen spel eller annan applikation, dessa indelas sedan ännu i flera olika kategorier. Innehålls klassificeringen tyder på ”åldersgränsen” för applikationen med fem olika nivåer. Sedan kan man ladda upp applikationen och hela paketet sparas som en skiss som inte ännu är tillgänglig för allmänheten. Sista steget är att kolla igenom allt material man laddat upp och alla konfigurationer. Efter det kan man trycka på en ”Publish” knapp för att göra applikationen tillgänglig för allmänheten. (Publishing Overview)

2.2.4 Utdelning via epost

Snabbaste och enklaste sättet att dela ut en applikation är via epost. Det räcker att kompilera applikationen till en apk-fil som beskrivits tidigare och därefter sätta det som bilaga till ett vanligt epost som skickas till användaren som man vill ge den åt. När användaren öppnar eposten med sin Android-enhet identifierar operativsystemet bilagan och öppnar en ”Installera nu”-knapp åt användaren. För att det skall fungera förutsätts att användaren har tillåtit installation av applikationer från oidentifierade källor på sin enhet och öppnat eposten med nativa Gmail-applikationen. Risken med att lansera applikationen via epost är att användarna kan lätt skicka vidare eposten och därmed dela ut applikationen till vem som helst.

2.2.5 Utdelning på webbsida

För att lansera applikationen på en egen webbsida måste man kompilera applikationen som normalt och sedan ladda upp den till webbsidan med en länk till filen. Webbsidan kan vara på en privat eller en enterprise server, det har ingen skillnad från applikationens synvinkel vilken som används. Användaren kan sedan öppna länken med sin Android-enhet som automatiskt laddar ner och installerar applikationen förutsatt att inställningarna är konfigurerade på samma sätt som i utdelning via epost.

3 UTVECKLINGTEKNOLOGIER

I detta kapitel presenteras olika verktyg och teknologier som används vid utvecklingen av den utvecklade applikationen och försöker förklara sammankopplingen mellan dessa.

3.1 Android

Android är ett Linux-baserat operativsystem med öppen källkod främst för mobiltelefoner och pekplattor som första gången lanserades 2007 av Google. I februari 2012 hade totala antalet aktiva användare överstigit 300 miljoner och numera aktiveras mer än en miljon nya Android-enheter dagligen.(Android, the world's most popular mobile platform) När detta arbete skrevs var den nyaste versionen av operativsystemet 4.4 som har projektnamnet "KitKat".

Google och Open Handset Alliance erbjuder en Software Development Kit(SDK) gratis till allmänheten. Den innehåller all information, API(Application Programming Interface) regeluppsättningar och verktyg för att göra applikationer med Java för Android. Systemet är utvecklat att vara plattformsoberoende vilket innebär att applikationerna kan lätt användas på flera olika fysiska enheter med avvikande hårdvara.

Det finns också en Native Development Kit(NDK) som ger utvecklare tillgång till nativ kod. NDK:n är vanligen inte plattformoberoende men kan däremot vara effektivare i vissa fall. (The Android OS)

3.2 NFC

NFC står för "Near Field Communication" och är en trådlös kommunikationsteknologi med låg bandbredd mellan två enheter med NFC-funktionalitet. Kommunikationen fungerar med frekvensen 13,56 MHz som också används av RFID(Radio Frequency Identification). Till skillnad från RFID som har en lång räckvidd har NFC en kort räckvidd och fungerar endast på mycket kort avstånd. NFC:s korta räckvidd anses göra data överföringen mer pålitlig i och med att båda parterna måste hämta enheterna mycket nära varandra för att kommunikationen skall ske. *(Coskun, Ok och Ozdenizci 2011 s.1)*

NFC teknologin möjliggör kommunikation mellan en mobiltelefon med NFC-möjlighet och en annan mobiltelefon med samma möjlighet, en NFC-läsare eller med ett elektroniskt kretskort med NFC-teknologi som kallas för NFC-tag. Möjliga framtida och redan nuvarande teknologier som kan utnyttja NFC-teknologin är elektroniska betalningar, identifiering, tillgångskontroll, innehålls fördelning, data- och penningtransaktioner och sociala tjänster.*(Coskun, Ok och Ozdenizci 2011 s.1)*

3.3 SQL

Structured Query Language(SQL) är ett programmeringsspråk för hantering av data i relationsdatabaser. Det utvecklades av IBM på 1970-talet för att hantera data i deras nya databassystem. År 1979 lanserade Relational Software, nuvarande Oracle, en kommersiell modifierad version av SQL som kallades för Oracle V2.

3.3.1 Allmänt om SQL

Numera är SQL en certifierad standard av både ANSI(American National Standards Institute) och ISO(International Organization for Standardization) som erbjuder många funktionaliteter såsom stöd för XML(Extensible Markup Language), reguljära uttrycks matchande, rekursiva sökningar och mycket flera. För att SQL standarden är så bred har det lett till att flera av de stora databas mjukvarorna som t.ex. MySQL och PostgreSQL

inte stöder hela standarden. Dessutom är hanteringen av datalagring och indexering inte tillräckligt bra definierad vilket leder till att de olika databas mjukvarorna är sällan kompatibla även om de alla baserar sig på SQL. (*SQL (Structured Query Language)*)

Fastän alla olika versioner av SQL inte stöder hela standarden, stöder de alla ändå de viktigaste funktionerna som är SELECT, UPDATE, DELETE, INSERT och WHERE. Av dessa funktioner byggs sedan SQL-frågor vilka skickas från en klient till en server där databasen finns. Servern kan sedan på basis av frågan räkna ut ett svar och returnera det till klienten. Det möjliggör grunderna till både lagring, hantering och sökning av data. (*SQL (Structured Query Language)*)

3.3.2 SQLite

SQLite är ett bibliotek som implementerar en självständig och serverlös SQL databas i form av en drivrutin. All kod för SQLite tillhör allmän domän, vilket innebär att den är gratis att använda för både kommersiellt och privat användning. Den används exempelvis av många stora företag och projekt som Google, Firefox webbläsaren, Dropbox och Skype. (*About SQLite*)

SQLite behöver inte en separat server process, utan istället läses och skrivs data direkt till vanliga diskfiler. Det innebär att SQL databasen med alla tabeller och funktionalitet finns lagrat lokalt på enheten i en fil på disksystemet. Filformaten för databasen fungerar på flera plattformar som 32-bit och 64-bit, samt på big-endian och little-endian arkitekturer. (*About SQLite*)

Biblioteket med all funktionalitet kan vara mindre än 350KB i storlek, beroende på plattformen och kompilatorns optimeringsinställningar. Den lilla storleken gör SQLite till ett populärt alternativ för enheter med begränsat minne som t.ex. mobiltelefoner och MP3-spelare. Även om databasen fungerar med mycket lite minne, ökar hastigheten av funktionaliteten ifall databasen har mera minne till sitt förfogande. (*About SQLite*)

3.4 Eclipse

Eclipse är ett samfund för öppen källkod som fokuserar på att erbjuda en omfattande utvecklingsplattform och applikationsramverk för mjukvaruutveckling. Eclipse-projektet är ursprungligen utgivet av IBM år 2001 till ~~öppna källkods~~ samfundet samtidigt som IBM grundade Eclipse-konsortiet. Konsortiet bestod av IBM:s partners och av konkurrenter. Numera förknippas konsortiet med Eclipse IDE(Integrated Development Environment) som är plattformen som IBM stödde utvecklingen av med olika bidrag samt kodningsläger. (*Eclipse IDE project resources*)

Eclipse IDE:n kan utvidgas med olika mjukvarukomponenter som kallas för plug-ins. För Android-utveckling finns ADT(Android Development Tools)-plugin. Den erbjuder ett grafiskt användargränssnitt för funktioner från SDK-verktygen och ett verktyg för skissning och utveckling av ett användargränssnitt till applikationen. Eftersom det är en plugin har den förutom sin egna funktionalitet också all funktionalitet som Eclipse IDE:n erbjuder till Java-utveckling. (*Android Developer Tools*)

3.5 Java

Java är ett objektorienterat högnivå programmeringsspråk som används på över 50 miljoner datorer och miljarder av apparaturer runt världen.

Java plattformen består av två huvudkomponenter, Java Application Programming Interface(API) som är en regeluppsättning av Java kommandorader och av Java Virtual Machine (JVM) som tolkar Java-koden till maskinspråk. (*The Java Programming Language and the Java Platform*)

Java är ett plattformsoberoende programmeringsspråk som fungerar genom att källkoden omvandlas i en kompilare till en bytekod som är plattformsoberoende. Det är JVM som är plattformspecifik som sedan tolkar och kör bytekoden vilket innebär att samma bytekod kan köras på flera olika maskiner. I programmeringsspråk som C eller C++ som körs nativt under operativsystemet bör den kompilerade koden innehålla platt-

formspecifik data, vilket innebär att alla plattformar måste ha egen kompilerad kod. I och med att Java kod tolkas när den körs är program skrivna i java något långsammare än nativt-kompilerade programmeringsspråk. (Greanier 2004 s.5)

I allmänhet görs applikationer för Android med Java men det finns även verktyg för utvecklandet av applikationer med C eller C++. I detta arbete används Java, programmeringsspråken C och C++ behandlas inte. (Android(Operating System))

4 UTVECKLINGSPROCESSEN

I det här kapitlet presenteras applikationen som utvecklades för projektet. I kapitlet presenteras själva applikationen, dess syfte och vissa utvecklingslösningar och problem som uppkom i utvecklingen.

4.1 Planering och kravställning

Applikationens syfte var att hantera arbetstiderna på en arbetsplats på ett centralt, simpelt och billigt sätt. För detta planerades en ”stämpelkorts” applikation där arbetarna kunde logga när de kom till arbetet, gick på lunch och när de lämnade arbetsplatsen. Loggningarna skulle sedan sparas i en databas därifrån man kunde söka arbetarnas arbetstimmar enligt behov. Vid utvecklingen användes en Google Nexus 7-pekplatta främst för dess relativt låga pris och för att den har NFC-funktionalitet. Därmed är applikationen utvecklad att fungera bäst på denna enhet.

För att unikt kunna identifiera de olika användarna med så lite inmatning av användaren som möjligt beslöts det att NFC-teknologin används. Genom att alla användare har egna unika NFC-taggar kunde en användare genom att visa taggen till enheten identifieras med ett unikt identifikationsnummer. Det möjliggör att användare kan lätt med endast två inmatningar, visande av NFC och val av aktivitet, använda applikationen snabbt och utan besvär. Med användning av NFC-teknologin måste applikationen också ha hante-

ring av användare och dess NFC-taggar som innebär skrivande av nya taggar, modifierande och radering av användare.

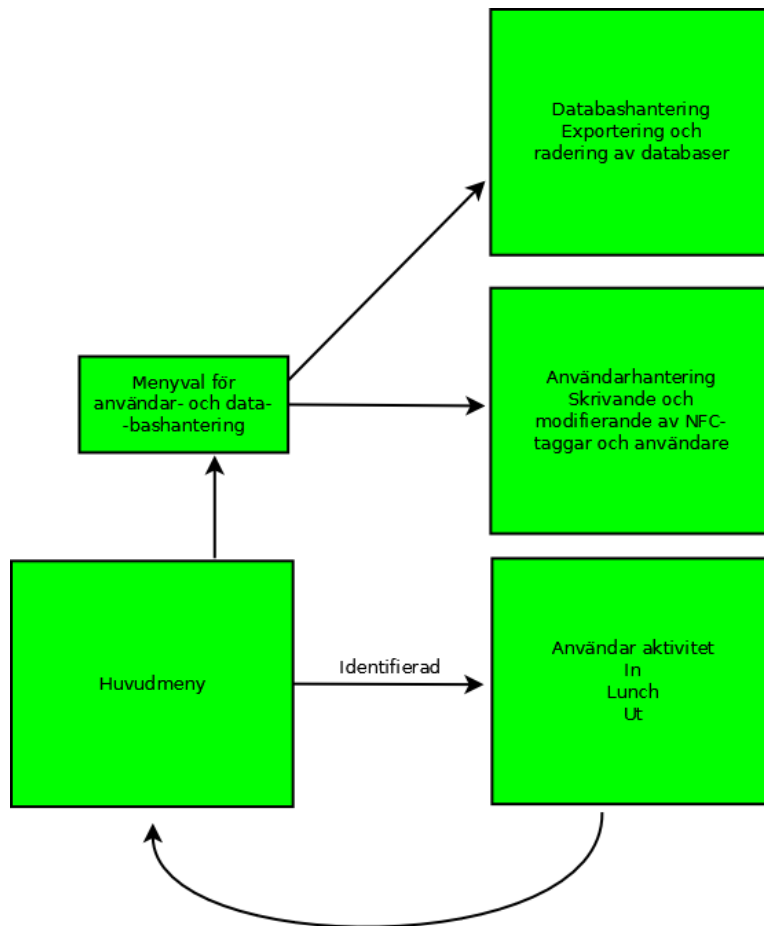
Eftersom Android systemet stöder SQLite-databasen beslöts det att den används istället för en extern databas över t.ex. nätanslutning. Det medförde att applikationen behöver ett sätt att kunna exportera databaserna från enheten till ett allmänt läsbart format. Databaserna som behövdes var en för användare och en för användargenererad data där aktivitets loggningarna sparas.

Applikationens kravställning blev därmed:

- Användare kan identifieras av applikationen med NFC-taggar
- Användare kan göra loggningar
- Användaraktiviteterna sparas i databas
- Användarhantering i databas
- Skrivande och modifierande av NFC-taggar
- Exportering av databaserna

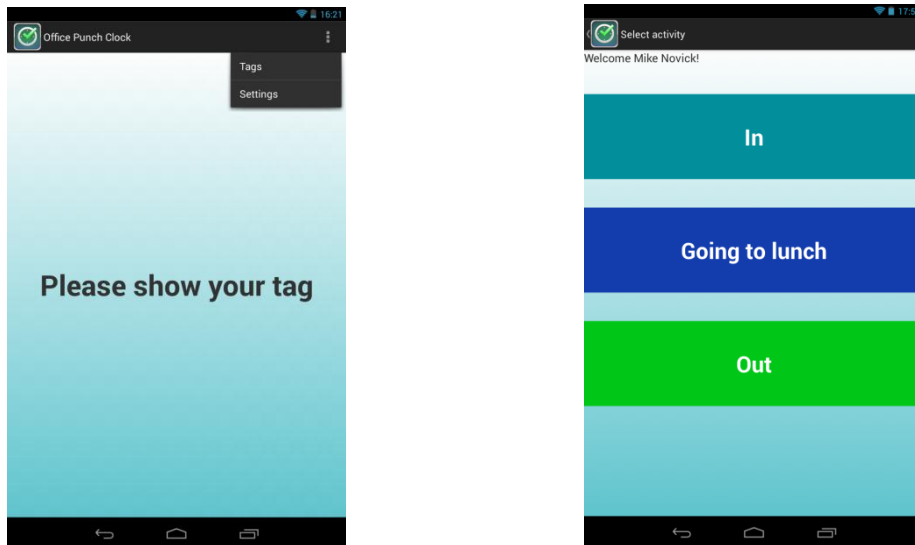
4.2 Användargränssnittet

Användargränssnittet skulle uppfylla kraven för så lite användarinmatning som möjligt med god användbarhet som inte behöver förklaras åt användarna. Nedan i Figur 2 är en presentation av hur användargränssnittet fungerar i applikationen.



Figur 2. Diagram över applikationens användargränssnitt

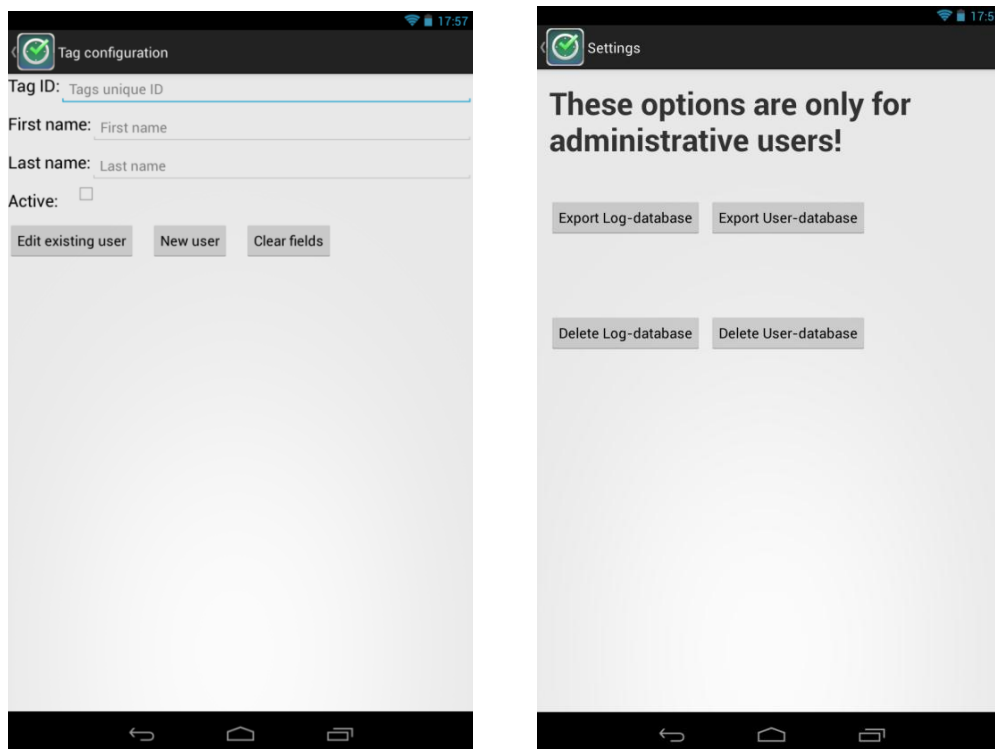
När applikationen startas öppnas ”Huvudmeny”-vyn där användaren uppmanas visa sin NFC-tag till enheten. Dessutom kan man öppna en rullgardinsmeny där användaren kan välja mellan vyer för användar-och databashantering. Nedan i Figur 3 är en skärmbild från applikationens huvudmeny med rullgardinsmenyn nerdragen.



Figur 3. Huvudmeny och användaraktivitet vyn av applikationen

Ifall användaren visar sin rätt konfigurerade och aktiva NFC-tagga vid huvudmenyn, öppnas "Användaraktivitet"-vyn där det visas tre knappar med texterna; In, Going to Lunch-eller Back from lunch och Out, som kan ses i figur 3. Lunch-knappens text varierar beroende på om användaren under samma dag har loggat till lunch, då visas texten Back from lunch och vice versa. När användaren trycker på en av knapparna returnerar applikationen till huvudmenyn.

Från huvudmenyn kan användare öppna en rullgardinsmeny med två knappar; "Tags" och "Settings". Om användaren väljer "Tags"-knappen öppnas användarhanterings-vyn, som kan ses nedan i figur 5, där användare kan göra nya användare och NFC-taggar och editera existerande användare.



Figur 4. Användar-och databashanteringsvyn från applikationen

Om användaren från huvudvyn öppnar rullgardinsmenyn och väljer ”Settings” öppnas databashanterings-vyn, som kan ses nedan i figur 5. I Databashanterings-vyn kan användaren välja att exportera databasernas data till en csv-fil på enhetens lokala minne. Eller också välja att radera all data från databaserna vilka innehåller alla loggningar gjorda av användare och all användardata.

4.3 Lösning av kravställningarna

I det här kapitlet går vi igenom hur de olika kravställningarna förverkligades i applikationen. I kapitlet går inte all kod bakom applikationen igenom utan endast de stora funktionerna som utfyllde kravställningarna.

4.3.1 Användarhantering i databas

I applikationen finns två lokala SQLite databaser. Den ena innehåller användarinformat-
ion och den andra alla loggningar som användarna gjort. Nedan i figur 7 ses strukturen
för användardatabasen med exempelinformation över en användare. I första kolumnen
finns nummerkoden som används för att identifiera användare, den används inom data-
basen som primärnyckel. I andra kolumnen är användarens förnamn och i tredje använ-
darens efternamn. I fjärde och sista kolumnen anmäls ifall användaren är aktiv vilket
skulle tydas med en "1" och som i exempelfallet med en "0" ifall användaren inte är satt
som aktiv.

Id(Integer Primary Key)	firstName(Text)	lastName(Text)	Active(Integer)
303010	Chloe	O'Brian	0

Figur 5. Användardatabas exempel

All databashantering sker i en klass som innehåller funktioner för byggandet av databa-
serna ifall de inte finns från tidigare och dessutom alla funktioner för hantering av data i
databaserna. Funktionerna som är viktiga för att lösa kravställningen över användarhan-
tering är att göra en ny användare, editera existerande och söka information över en exi-
sterande.

4.3.2 Skrivande och modifierande av NFC-taggar

Användare är identifierade med en unik nummerkod med 6 slumpmässiga numror.
Nummerkoden skrivs sedan på NFC-taggen varifrån den kan bli avläst av applikationen
och därmed användas för att identifiera enskilda användare.

När användaren öppnar vyn för användarhantering, som kan ses i figur 5, startas en
klass med en intent-lyssnare. Intent-lyssnare är en passiv datastruktur som innehåller en
abstrakt beskrivning av en funktion som skall köras(Intent), och när en inträffar kollar
den ifall det var en NFC-aktiverad intent. Dessutom innehåller klassen variabler som
kontrollerar ifall användaren vill skriva nya taggar eller läsa redan existerande taggar
och söka deras information. Variablerna är boolean-värden som ändras beroende på vil-

ken av knapparna; "Edit existing user" eller "New user" användaren tryckt på. Ifall ingendera av knapparna är tryckta hanteras inte den lästa taggen i intent-lyssnar funktionen. Om användaren trycker på "Edit existing user"-knappen öppnas ett dialog fönster som uppmanar användaren att visa taggen av användaren som skall editeras. Då ändras boolean värdena så att när intent-funktionen läst en taggs nummerkod, anropas funktionen för att söka existerande användare från databasklassen vilket returnerar informationen för användaren till användarhanteringsvyn. För att skapa en ny användare och tagg skall användaren först mata in informationen till de lediga lådorna och därefter trycka på "New user"-knappen. Då öppnas ett dialog fönster som uppmanar att visa en ledig tagg. När intent-funktionen läser taggen görs en ny slumpmässig nummerkod som sedan kollas mot databasen för att undvika likadanna värden. Därefter skickas den nya slumpmässiga nummerkoden och de inmatade värdena till funktionen i databasklassen som gör en ny användare till databasen.

4.3.3 "Användare kan identifieras av applikationen med NFC-taggar"

När applikationen startas körs en klass med en intent-lyssnare. Sedan läser en funktion NFC-taggen och därefter läser den datan och sparar det i en string variabel. String variabeln kollas sedan med en funktion i databasklassen som kollar ifall variabelns nummerkod finns i användardatabasen och ifall denna användare är aktiv. Ifall funktionen returnerar ett nummer högre än 0, så är taggen giltig och applikationen skickar nummersträngen vidare till nästa funktion. Om nummerkoden inte fanns i databasen eller om användaren inte var aktiv visas texten "Not a valid tag!" åt användaren och applikationen gör ingenting vidare. Nedan i figur 7 ses funktionen "onNewIntent".

```

public void onNewIntent(Intent intent1)
{

    if(NfcAdapter.ACTION_TAG_DISCOVERED.equals(intent1.getAction()))
    {
        Context contextNew = this;
        DatabaseHandler dbh = new DatabaseHandler(contextNew);

        Tag myTag = (Tag) intent1.getParcelableExtra(NfcAdapter.EXTRA_TAG);
        String result = tdh.readTag(intent1);

        int validCount = dbh.checkPersonActive(Integer.parseInt(result));

        if(validCount > 0)
        {
            sendTagInfo(result);
        }
        else
        {
            Toast.makeText(getApplicationContext(), "Not a valid tag!", Toast.LENGTH_LONG).show();
        }
    }
}

```

Figur 6. Funktionen "onNewIntent"

4.3.4 Användare kan logga in, till lunch och ut

Ifall användaren visat en giltig NFC-tag i startvyn av applikationen, som beskrevs i föregående avsnitt, körs en klass som ses av användaren som användaraktivitets-vyn. När klassen startades från huvudvyn sätts den lästa NFC-taggens identifieringsnummer som intent, och sparas sedan i en variabel den startade klassen. Variabeln används för att identifiera användaren och skriva nummerkoden till loggdatabasen i samband med den gjorda loggen.

När användaren trycker på en av de tre knapparna kallas en funktion i databasklassen med användarens nummerkod och beroende på vilken knapp som trycktes en teckensträng som indikerar ifall användaren loggade in, till lunch, tillbaka från lunch eller ut. Lunch-knappen varierar texten och därmed texten som skrivs i databasen beroende på ifall användaren redan gjort en loggning under samma dag som innehöll ordet "lunch". En funktionen gör därefter en ny rad i loggdatabasen, vars exempel rad kan ses nedan i figur 9. I första kolumnen är ett identifikationsnummer för enskilda rader, detta nummer insätts automatiskt av databasen och är alltid ett värde större än det tidigare. I andra kolumnen är den mottagna användarens identifikationsnummer. I tredje kolumnen är da-

tum och tid när raden och därmed loggen gjordes, den skrivs också automatiskt av databasen. I fjärde kolumnen är det mottagna textsträngen av användaren som tyder på vilken typ avloggning det är fråga om.

LogID(Integer Primary key, Autoincrement)	TagID(Integer)	Date(Datetime default current_timestamp)	Type(Text)
40	303010	2013-04-1 16:26:55	in

Figur 7. Loggdatabas exempel

När användaren tryckt på en av knapparna returnerar applikationen till huvudvyn.

4.3.5 Exportering av databaserna

När användaren öppnar databashanterings-vyn körs en klass med två lyssnare för de två olika knapparna för exportering av de olika databaserna. När en av knapparna trycks kallas funktionen "createCSVpersons" eller "createCSV" beroende på vilken av knapparna som trycktes. Båda funktionerna fungerar mycket liknande men läser olika data från databasen. De kallar sedan på funktioner från databashanterings-klassen som läser alla rader från ifrågavarande tabell till en lista som sedan returneras. Listan består av raderna från databasen där varje kolumns värde är separerad från varandra med ett ";"-tecken. Den mottagna listan går sedan igenom radvis där varje rad skrivs i en fil med en FileOutputStream-funktion. Resultatet är en .csv-fil som sparas på enhetens lokala minne, där alla värden från databasen är separerade med ";"-tecken på sina egna rader precis som de är i databasen.

5 LANSERINGSPROCESSEN

Lanseringen av applikationen bestod av olika delmoment som presenteras i detta kapitel.

5.1 Förberedningar

För att förbereda applikationen för publicering bör källkoden städas och optimeras. Man började med att gå igenom alla källkodsfiler och raderade ut alla avlusningsfunktioner. I varje källkodsfil söktes med Eclipses inbyggda sökfunktion orden "System.out.println" och vid hittade ställen raderades koden. Samma gjordes även för sökordet "Toast" där alla meddelanden som var använda för avlusning raderades medan de viktiga som ger användaren information lämnades kvar.

Applikationen hade en färdig ikon för startande av applikationen och dessutom var den synlig inne i applikationen vid övre kanten. En ny ikon gjordes och överfogades till applikationen med olika dimensioner för olika skärmdensiteter som ses i Figur 1. I manifest filen uppdaterades värdena för <android:icon> och <android:label> för att stöda den nya ikonerna. Med samma kollades att <android:versionCode> och <android:versionName> attributerna fanns och var korrekta.

Tabell 1. Förhållande mellan ikon storlek och skärmdensitet(Supporting Multiple Screens)

Ikon storlek i pixel	Skärm densitet
36x36	Låg densitet(120), ldpi
48x48	Medel densitet(160), mdpi
72x72	Hög densitet(240), hdpi
96x96	Extra hög densitet(320), xhdpi

Dessutom gjordes en ikon med storleken 512*512 pixel som krävs för Google Play webbsidorna.

Till applikationen gjordes en ny klass som visar en dialog första gången applikationen körs eller ifall den körs efter en uppdatering. I dialogen visas applikationens EULA och dessutom nyaste uppdateringarna i applikationen ifall det är aktuellt. Användaren kan stänga dialogen med att acceptera EULA:n(Bilaga 1) eller alternativt körs inte applikationen.

Några skärmbilder skapades med en Nexus 7 pekplatta med pekplattans inbyggda funktioner. Dessutom skrevs en reklamtext som beskriver applikationens funktion och syfte för att presenteras på Google Play.

Applikationen kompilerades för att se att ikonerna och EULA-dialogen fungerade rätt och att funktionaliteten av applikationen inte hade förändrats och konstaterades att allting fungerade som det skulle.

5.2 Kompilering och signering

I Eclipse ADT valdes projektet för exportering. Programmet frågade ifall man hade färdiga nyckelpar och om inte skapades nya. För provningens skull kördes hela processen där nya nyckelpar skapades och som resultat kompilerade, signerade och optimerade applikationen till en apk-fil till ett valt ställe på datorn. Vid skapande av nycklarna måste man fylla i en beskrivning av nyckeln samt ett lösenord. Nyckeln behöver inte signeras av en utomstående CA(Certificate Author).

5.3 Google Play lansering

För att ladda upp applikationer till Google Play måste man ha ett Google Play publisher-konto.

Ett nytt Google-konto skapades vilket inte krävde annat än kontaktinformation och en epost-address. För publisher-kontot krävdes information för applikationens utgivare och ett namn för utgivaren som syns under alla publicerade applikationer. För att få kontot

måste man betala en engångsavgift på 25\$ vilket gjordes med kreditkort där betalningens verifieringsprocess anmäldes räckta upp till 48 timmar. Eftersom applikationen sattes som en gratis applikation behövdes inte ett Google Checkout merchant-konto. Google Checkout Merchant-kontot skulle innehålla bankinformation för betalningar och information över beskattning.

Medan verifieringsprocessen var igång kunde man ladda upp applikationens apk-fil, skärmbilder och reklamtexter färdigt till Google Play som en "skiss" vilket innebär att applikationen inte blir allmänt tillgänglig förrän användaren väljer det. För att få applikationen godkänd för allmän utgivning måste man förutom ladda upp diverse filer också godkänna två kontrakt som där man övertygade att applikationen fyller Androids innehållsriktlinjer och att applikationen fyller Förenta Staternas export lagar gällande mjukvaror. Dessutom måste länder där applikationen skall vara tillgänglig väljas där alla 137 länder valdes.

När alla nödvändiga punkter var uppfyllda och betalningen var verifierad publicerades applikationen på Google Play vilket anmäldes kunna ta upp till några timmar. Efter ungefär tre timmar kunde applikationen ses på Google Play för allmänheten.

6 HUR ÖKA NERLADDNINGAR

I detta kapitel går igenom möjliga alternativ för att få mera nerladdningar för den lanserade applikationen. Metoderna provades inte i arbetet utan presenteras endast som teoretiskt möjliga.

6.1 Metoder

På Google Play kan användare värdera nerladdade applikationer på en skala från 1-5 och tillägga en kort beskrivelse av deras upplevelse av applikationen. Denna värdering är en av de största faktorerna i applikationens rangordning i Google Play. Rangon och

antalet nerladdningar påverkar hur applikationen visas på olika listningar och sökningar inom Google Play. Det är också en av de viktiga sakerna redaktörerna tittar på när de väljer applikationer för olika marknadsförnings kampanjer.(Visibility for your Apps) ^{ref}

Eftersom Google inte berättar hur deras sorterings algoritmer fungerar, är källorna för följande metoder tagna från olika bloggar där utvecklare har provat sig fram till ifrågasvarande metoder. Flera olika bloggar har nämnt liknande metoder och spekulerat på samma vis över hur Googles sorterings algoritmer fungerar, därmed antas att denna information kan vara indikativ.

6.1.1 Applikationen

Den viktigaste delfaktorn i att få mycket nerladdningar för en applikation är att bygga en bra applikation. Applikationen kan t.ex. vara en som gör någonting helt nytt eller har en helt ny vy på en redan existerande applikation. Den borde dock inte vara en direkt kopia av någon av de redan existerande populära applikationerna på applikationsmarknaden. En bra applikation beror inte endast på en bra idé utan även på uppbyggnaden, verkställandet, design, användarerfarenhet och även detaljer som en bra ikon. (You built it but they didn't come: 8 tricks for marketing your mobile app, 2012)

För att verkställa en bra applikation kan man följa Googles egna riktlinjer vilka presenterades i kapitel 3. Som Google också rekommenderar bör man satsa på en bra reklamtext för applikationen. Eftersom sökfunktionen på Google Play söker igenom denna text borde man inkludera så kallade nyckelord i texten, enligt vissa källor borde nyckelordet uppkomma 5 gånger i texten. Nyckelorden är ord som beskriver funktionaliteten eller syftet med applikationen. Nyckelordet borde även uppkomma i applikationens namn.(Optimize Your Google Play Store App Details Page)

6.1.2 Användarvärderingar

Som redan tidigare nämnts är användarvärderingar en viktig delfaktor i rangen för applikationen. Dessutom ser andra användare värderingarna vilket betyder att en bra värdering för applikationen kan ge användaren en bra bild över applikationen och därmed ladda applikationen. Även här ses betydelsen av en bra applikation, ifall applikationen inte är bra får den knappast bra värderingar. För att få värderingar kan man rekommendera applikationen till sina bekanta och arbetskollegor och be dem att värdera applikationen. För att få användarna och värdera applikationen kan man bygga in i applikationen en dialog som ber användaren att värdera den. (You built it but they didn't come: 8 tricks for marketing your mobile app, 2012)

6.1.3 Marknadsföring

Man kan stöda bilden över en bra applikation med att göra en egen webbsida åt applikationen som innehåller t.ex. bilder eller videon om applikationen. Man kan även nämna och marknadsföra applikationen på sociala medier som Facebook, Twitter, Google+ och Youtube via egna konton och dessutom be bekanta att dela dem vidare. Man kan även blogga om applikationen och delta i diskussioner på utvecklar forums för att få mera synlighet och kritik för applikationen. Samt kan man betala åt värderingssidor som sedan publicerar och värderar applikationen på sina sidor. Därmed får applikationen mycket synlighet och möjliga nerladdningar. Det finns flera av dessa sidor och därmed bör man studera noggrant att sidan är legitim förrän man gör någon betalning. Alla sidor som länkar till applikationen ökar på Googles sidrang vilket i sin tur kan höja sidans rang i sökningar.

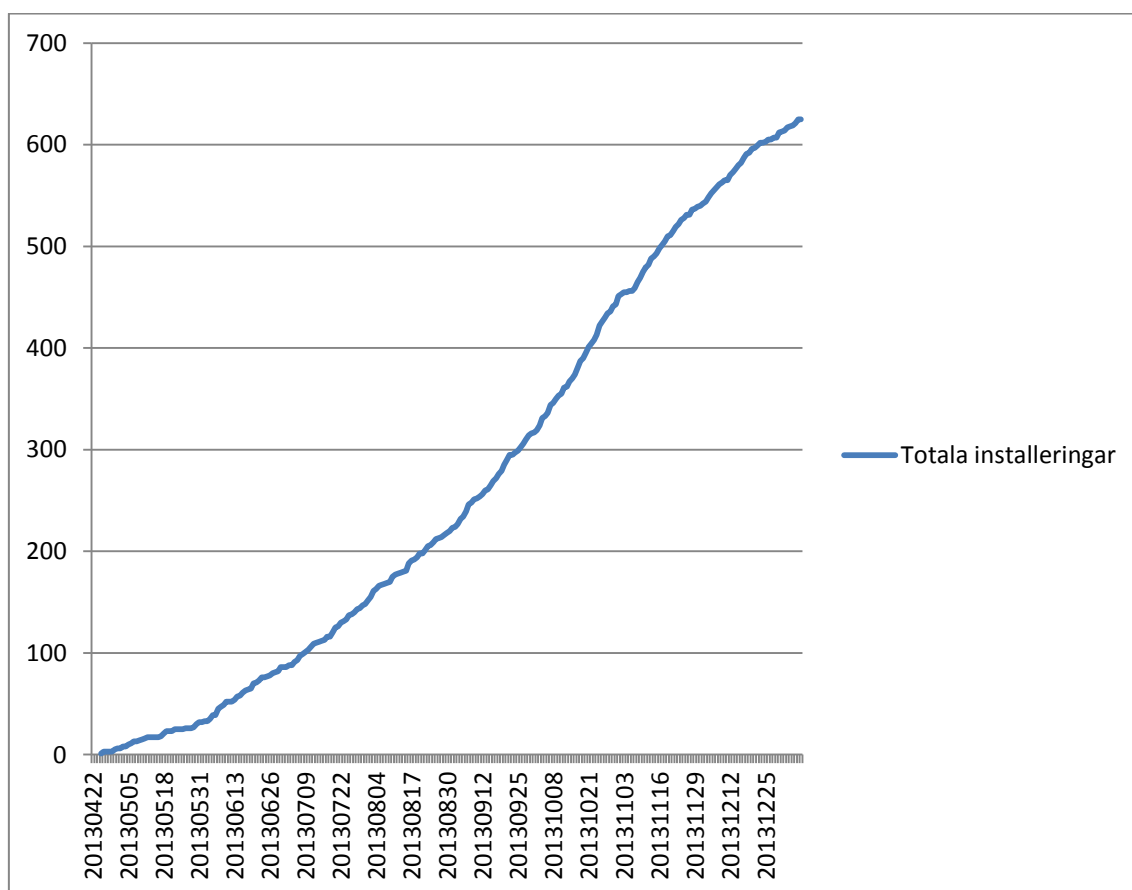
6.2 Process

Eftersom arbetet inte skall gå in på marknadsföringen av applikationen har dessa metoder inte används. När arbetet lanserades sattes det mycket tanke på reklamationstexten, ikonerna och redan i tidigare skede användbarheten och funktionerna av applikationen. Därmed är det svårt att göra några förbättringar i ett senare skede och försöka få pålitlig

data över resultaten. Applikationen rekommenderades dock till bekanta som även gav applikationen några värderingar.

6.3 Resultat och analys

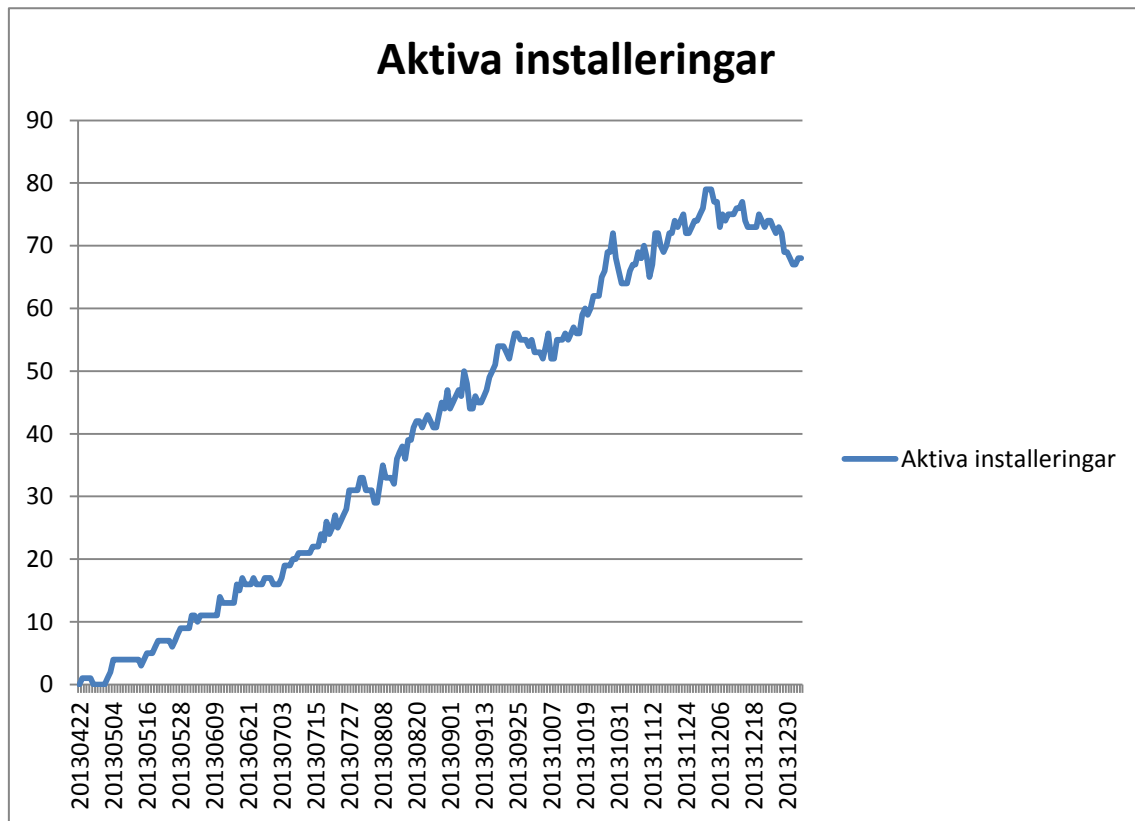
Eftersom inga direkta metoder har använts för att öka nerladdningarna efter lanseringen kan man inte analysera två olika grafer som skulle ha varit tagna före användningen av presenterade metoderna samt en graf efter att metoderna för flera nerladdningar verkställts. Istället presenteras här hur nerladdningarna för applikationen gått i allmänhet. Nedan i figur 10 ses antalet nerladdningar för applikationen sen lanseringen till 4.1.2014.



Figur 8. Totala installeringar

Figuren visar totala antalet installeringar, som inte alltså innehåller data över hur många som senare raderat applikationen. Det senaste antalet installeringar är 625. Man kan se att antalet installeringar har varit mycket konstant under livstiden av applikationen. I

figur 11 nedan kan man se en graf över antalet aktiva installeringar, där alltså endast räknas hur många som under just nu har applikationen installerad på en enhet.



Figur 9. Aktiva installeringar

På samma sätt som i totala installeringar har även här antalet ökat i en konstant takt, förutom i slutet där antalet börjat minska. Som slutsats kan man säga att lanseringen och utvecklingen varit lyckat med tanke på att antalet användare ökat i konstant takt och applikationen har en värdering på 4,33 med tre totala röster.

7 SLUTSATSER

Målsättningen med arbetet var att utveckla en Android-applikation, lansera den samt att studera ifall det var möjligt att öka antalet nerladdningar för den lanserade applikationen. Utvecklingen av applikationen gick bra med Eclipse ADT och med en pekplatta inkopplad för avlusning redan från början av utvecklingen. Det hjälpte mycket i att se hur applikationen fungerar i praktiken och var till stor nytta i utvecklandet av NFC-funktionaliteten. Likaså gick lanseringen av applikationen smidigt med verktygen som Android erbjuder åt utvecklarna.

Under arbetet har jag lärt mig mycket om hur NFC-teknologin fungerar i praktiken och ur den kan implementeras i applikationer.

Den utvecklade applikationen har fått mycket mera nerladdningar än jag förväntade mig vilket är mycket positivt. Den har också fått mycket bra kritik och förfrågningar för ny funktionalitet som kan implementeras senare. Det lärde mig också att även om en applikation är lanserad är den inte nödvändigtvis helt och hållet ”färdig” ännu.

Metoderna för mera nerladdningar skulle man kunna ha provat med att lansera en version av applikationen med en dålig ikon och dålig reklamtext och sedan stegvis implementerat de presenterade metoderna och därmed kunnat få data där man kunde se skillnader i nerladdningstakten och kunna konstatera ifall metoderna fungerade.

Syftet med arbetet var att klargöra lanseringen av en Android-applikation vilket jag tycker arbetet gör i och med att den presenterar ett praktiskt exempel där lanseringen lyckades. En ny utvecklare kunde nu läsa arbetet och veta vad man kan förvänta sig och hur man kan förbereda sig för lanseringen redan vid utvecklinsskedet.

KÄLLOR

About SQLite

Tillgänglig: <http://www.sqlite.org/about.html>

Hämtad 6.5.2013

Android Developer Tools

Tillgänglig: <http://developer.android.com/tools/help/adt.html>

Hämtad 6.5.2013

Android the world's most popular mobile platform.

Tillgänglig: <http://developer.android.com/about/index.html>

Hämtad 29.4.2013

Android(Operating System)

Tillgänglig: [http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))

Hämtad 13.11.2013

Application Licensing

Tillgänglig: <http://developer.android.com/google/play/licensing/index.html>

Hämtad 13.11.2013

Coskun Vedat, Ok Kerem, Ozdenizci Busra. 2011, Near Field Communication : From Theory to Practice (2nd Edition), 2uppl., Hoboken, NJ, USA: Wiley
Wiley, 390s.

Eclipse IDE project resources

Tillgänglig: <http://www.ibm.com/developerworks/opensource/top-projects/eclipse/eclipse-starthere.html>

Hämtad 6.5.2013

EULA

Tillgänglig: <http://www.webopedia.com/TERM/E/EULA.html>

Hämtad 13.5.2013

Graphic and Image Assets

Tillgänglig: <https://support.google.com/googleplay/android-developer/answer/1078870>

Hämtad 13.5.2013

Greanier, Todd. 2004, Java Foundations, Alameda, CA, USA: Sybex
Sybex, 363 s.

Intent

Tillgänglig: <http://developer.android.com/reference/android/content/Intent.html>
Hämtad 27.9.2013

Optimize Your Google Play Store App Details Page

Tillgänglig: <http://www.apptamin.com/blog/optimize-play-store-app/>
Sylvain Gauchet
Hämtad 6.1.2014

Preparing for Release

Tillgänglig: <http://developer.android.com/tools/publishing/preparing.html>
Hämtad 22.4.2013

Publishing Overview

Tillgänglig:
http://developer.android.com/tools/publishing/publishing_overview.html
Hämtad 9.5.2013

Signing your Applications

Tillgänglig: <http://developer.android.com/tools/publishing/app-signing.html#cert>
Hämtad 13.5.2013

SQL (Structured Query Language)

Tillgänglig: <http://www.ntchosting.com/databases/structured-query-language.html>
Hämtad 6.5.2013

Strömquist, Siv. 2006, Uppsatshandboken, 4 uppl., Uppsala: Hallgren & Fallgren
Studieförlag AB, 142 s.

Supporting Multiple Screens.

Tillgänglig:
http://developer.android.com/guide/practices/screens_support.html#testing
Hämtad 22.4.2013

Tech Specs

Tillgänglig: <http://www.google.com/nexus/7/specs/>
Hämtad 6.5.2013

The Android OS, Open Operating System System for Mobile Devices

Tillgänglig: <http://linux.about.com/od/mobiledevices/a/The-Android-Os.htm>
Hämtad 29.4.2013

The Java Programming Language and the Java Platform

Tillgänglig:
<http://www.oracle.com/technetwork/topics/newtojava/downloads/index.html>
Hämtad 29.4.2013

What To Test

Tillgänglig: http://developer.android.com/tools/testing/what_to_test.html

Hämtad 16.5.2013

Visibility for your Apps

Tillgänglig:

<http://developer.android.com/distribute/googleplay/about/visibility.html>

Hämtad 30.10.2013

You built it but they didn't come: 8 tricks for marketing your mobile app, 2012

Tillgänglig: <http://venturebeat.com/2012/05/30/marketing-your-new-app-ios-android/>

Hämtad 6.1.2014

Zipalign

Tillgänglig: <http://developer.android.com/tools/help/zipalign.html>

Hämtad 13.11.2013

BILAGOR

BILAGA 1.

Eula: The product is protected by Federal Copyright Law. Toni Malila retains the title to and ownership of the Product. You are licensed to use this Product on the following terms and conditions:

LICENSE - The licensee is defined as the individual or company utilizing the Software Product. Toni Malila hereby grants the licensee a nonexclusive license authorizing the licensee to use the enclosed Product on one computer at a time. The licensee is also permitted to distribute this product to one, and only one web server to host the Program. Please contact Toni Malila if you require additional licenses. Use of this product by more than one individual or by anyone other than the licensee terminates, without notification, this license and the right to use this product.

YOU MAY NOT: Distribute, rent, sub-license or otherwise make available to others the software or documentation or copies thereof, except as expressly permitted in this License without prior written consent from Toni Malila. In the case of an authorized transfer, the transferee must agree to be bound by the terms and conditions of this License Agreement.

RESTRICTIONS: - You may use this Product in your business as long as: The software serial number and user must be registered with Toni Malila in order to receive support or distribution rights. You may not remove any proprietary notices, labels, trademarks on the software or documentation. You may not modify, de-compile, disassemble, reverse engineer or translate the software.

US GOVERNMENT RESTRICTED RIGHTS - Use, duplication or disclosure by the United States Government is subject to restrictions as set forth under DFARS 252.227-7013 or in FARS 52.227-19 Commercial Computer Software - Restricted Rights.

TERM - You may terminate your License and this Agreement at anytime by destroying all copies of the Product and Product Documentation. They will also terminate automatically if you fail to comply with any term or condition in this Agreement.

LIMITED WARRANTY - This software and documentation are sold "as is" without any warranty as to their performance, merchantability or fitness for any particular purpose. The licensee assumes the entire risk as to the quality and performance of the software. Toni Malila warrants that the diskettes on which the Program is furnished will be free from any defects in materials. Exclusive remedy in the event of a defect is expressly limited to the replacement of diskettes. In no event shall Toni Malila or anyone else who has been involved in the creation, development, production, or delivery of this software be liable for any direct, incidental or consequential damages, such as, but not limited to, loss of anticipated profits, benefits, use, or data resulting from the use of this software, or arising out of any breach of warranty.